$u(x,t) = e^{\beta t} \sin \alpha x$

(a) For $u(0,t) = 0$, any α will do because $\sin(0 \cdot x) = \sin 0 = 0$.

For $u(L,t) = 0$, need $\sin \alpha L = 0$, so need $\alpha L = p\pi$, $p \in \mathbb{Z}$.

Answer: $\boxed{\alpha = \frac{p\pi}{L}, \ p \in \mathbb{Z}}$

(b) If $u(x,t) = e^{\beta t} \sin \frac{p\pi x}{L}$, then

$u_t = \beta u$ and

$u_{xx} = -\left(\frac{p\pi}{L}\right)^2 u$, so need

$$\beta \cancel{u} = D \cdot \left(-\left(\frac{p\pi}{L}\right)^2 \cancel{u} \right)$$

$$\boxed{\beta = -D\left(\frac{p\pi}{L}\right)^2}$$

So each function

$$u(x,t) = e^{-\frac{Dp^2\pi^2}{L^2} t} \sin \frac{p\pi x}{L}, \ p \in \mathbb{Z}$$

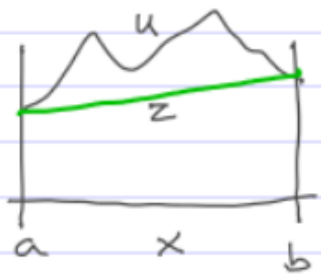
is a solution of the heat equation. All decay exponentially in time, and those with finer structure (larger $|p|$) decay faster.

$u_t = D u_{xx}, \quad a < x < b, \quad t > 0$

$$\left.\begin{aligned} u(a,t) &= \ell(t) \\ u(b,t) &= r(t) \end{aligned}\right\} \quad t > 0$$

Define $z(x,t)$ as the linear interpolant between $\ell(t)$ and $r(t)$ at each $t \geq 0$. That is

$$z(x,t) = \ell(t)\,\frac{b-x}{b-a} + r(t)\,\frac{x-a}{b-a}$$



And define

$$v(x,t) = u(x,t) - z(x,t)$$

Then

$$v(a,t) = u(a,t) - \left(\ell(t)\cdot 1 + r(t)\cdot 0\right)$$
$$= \ell(t) - \ell(t) = 0 \quad (\forall t)$$

and

$$v(b,t) = u(b,t) - \left(\ell(t)\cdot 0 + r(t)\cdot 1\right)$$
$$= r(t) - r(t) = 0 \quad (\forall t).$$

Additionally,

$$u(x,t) = v(x,t) + z(x,t), \text{ and}$$

$$u_t = v_t + z_t = v_t + \ell'(t)\,\frac{b-x}{b-a} + r'(t)\,\frac{x-a}{b-a}$$

$$u_{xx} = v_{xx} + z_{xx} = v_{xx} + 0$$

Therefore $u_t = D u_{xx}$ iff

$$v_t + \ell'(t)\frac{b-x}{b-a} + r'(t)\frac{x-a}{b-a} = D v_{xx}$$

or, to write it in another way

$$u_t - D u_{xx} = 0 \quad \text{iff}$$

$$v_t - D v_{xx} = -\ell'(t)\frac{b-x}{b-a} - r'(t)\frac{x-a}{b-a}$$

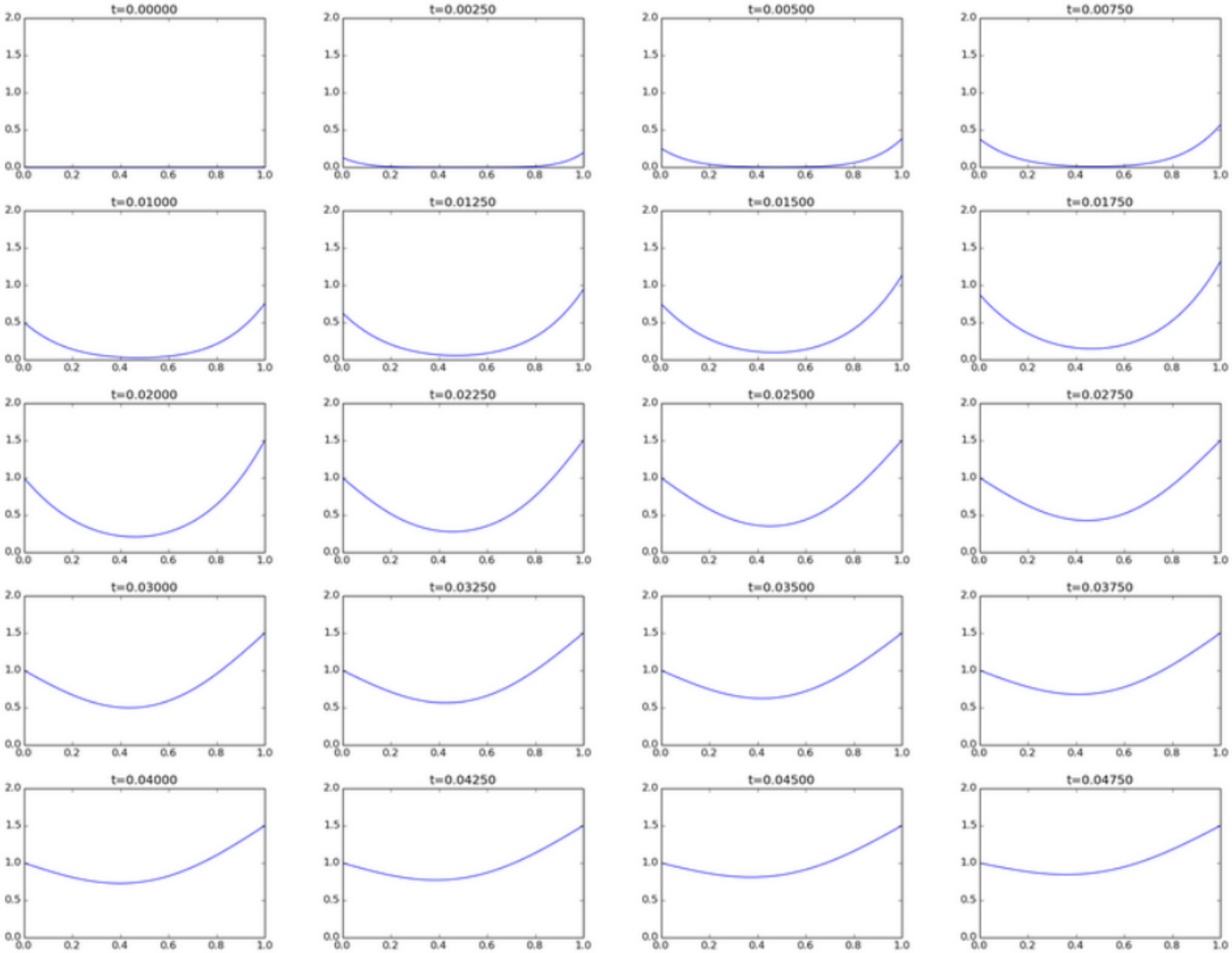
In conclusion, the original IBVP for u with <u>in</u>homogeneous BCs is equivalent to another heat equation IBVP for v, where the <u>PDE</u> has an inhomogeneous ("source") term but the <u>BCs</u> are homogeneous (i.e. $v(a,t) \equiv 0 \equiv v(b,t)$).

Thus we can consider homogeneous (zero) BCs without loss of generality.

3 The code :

```
from numpy import *
from pylab import *
from time import sleep

M = 50       # number of spatial subintervals
L = 1.
N = 2000  # number of time steps
T = 0.1
h = L/float(M)  # spatial grid spacing
k = T/float(N)  # time step
m = M - 1
D = 4.

# boundary conditions
tsteady = .02
rate = 50.
def l(t):
      if t<tsteady:
              return rate*t
      else:
              return rate*tsteady

def r(t):
      return 1.5*l(t)

sigma = D*k/h**2
print "sigma =",sigma

# initialization
x = linspace(0,L,m+2)
u = 0*x
u[0] = l(0.)
u[-1] = r(0.)

figure(figsize=(5,3))
ion()
plot([0,L],[0.,2.],'w') # set the plot scale
p,=plot(x,u,'b',linewidth=1)

for j in range(N):
      print j

      # calculation
      u[1:-1] += sigma*( u[0:-2] - 2.*u[1:-1] + u[2:]  )
      t = float((j+1)*k)
      u[0] = l(t)
      u[-1] = r(t)

      # graphics
      p.set_ydata(u)
      title('t='+'{:.5f}'.format(j*k))
      draw()
      #if j*k > 0.01: break
      if t>.05: break #abs(u).max()>8.: break
      if j%5==0:
              savefig('heat.hw4_'+str(j).zfill(5)+'.png')
ioff()
#show()
```

Results (see also the animation):



What happens when $\sigma = 0.512820512821$ - a little beyond the stability limit of 0.5: